



## Minimization of delay and collision with cross cube spanning tree in wireless sensor networks

Zhang, J., Xu, L., Tsai, P-W., & Lin, Z. (2019). Minimization of delay and collision with cross cube spanning tree in wireless sensor networks. *Wireless Networks*, 25, 1875–1893. <https://doi.org/10.1007/s11276-017-1653-4>

[Link to publication record in Ulster University Research Portal](#)

**Published in:**  
Wireless Networks

**Publication Status:**  
Published (in print/issue): 01/05/2019

**DOI:**  
[10.1007/s11276-017-1653-4](https://doi.org/10.1007/s11276-017-1653-4)

**Document Version**  
Author Accepted version

### General rights

Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

### Take down policy

The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [pure-support@ulster.ac.uk](mailto:pure-support@ulster.ac.uk).

# Minimization of Delay and Collision with Cross Cube Spanning Tree in Wireless Sensor Networks

Jing Zhang · Li Xu · Pei-Wei Tsai ·  
Zhiwei Lin

Received: date / Accepted: date

**Abstract** The Wireless Sensor Networks (WSNs) is a system containing the event detection and the data gathering abilities. The data gathering mechanism is the fundamental but important procedure in the WSN environment. The way of the data gathering majorly affects the efficiency of WSNs on retrieving data at the sink node. It is generally known that the clustering techniques are effective to reduce the energy consumption in the WSNs. However, the discussion on the packet collision and the transmission delay in the CDS based routing algorithms still remains vacant. The packet loss and the transmission delay may raise because of the collision and cause the negative impact of the WSNs performance. In addition, the transmission delay phenomenon in the WSNs may cause the inefficient result in the data gathering process. Unfortunately, it is usually neglected in the existing literature. To overcome the drawback of transmission delay and collision, a Cluster-based Converge Cast Tree (CCCT) structure is proposed in this paper. Furthermore, three major problems including the Minimal Connected Documenting Set (MCDS) problem, the delay aware CDS problem, and the CCCT construction problem are revealed in this study. Finally, the proposed algorithms are examined by the simulations. The simulation results indicate that the proposed CCCT struc-

---

J. Zhang (\* corresponding author)

School of Information Science and Engineering, Fujian University of Technology, and Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou, China, 350118.  
E-mail: jing165455@126.com

L. Xu

School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, China, 350007.

P.-W. Tsai

Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, Australia, 3122.

Z. W. Lin

School of Computing, Ulster University, Jordanstown, Northern Ireland, UK, BT370QB.

ture and algorithms outperform the existing approaches significantly in the realistic WSN environment.

**Keywords** Wireless Sensor Networks · Connected Documenting Set · Collision · Delay · Spanning Tree

## 1 Introduction

The inherent characteristics of the *Wireless Sensor Networks* (WSNs) provides two major abilities: the first is the sensing capacity for detecting the occurrence of the concerned events; and the second is the transmission competence of sending the detected data back to the sink node, which is commonly known as the data process center in the WSNs. Thus, it is widely used in many applications such as monitoring of the diverse environment and collecting data for the structural health diagnosis. The WSNs is composed of large amounts of resource-limited wireless sensor nodes[1]. These sensor nodes are usually deployed physically nearby the targets of interest in a close range to maximise the detection coverage. The data of the detected incidents are encapsulated in packets and are transmitted back to the sink node through the communication mechanisms embedded in the WSNs. Since the transmission of the data is accomplished via the data transmission channels, the packet collision problem always occurs when two or even more sensor nodes attempt to push the packet into the communication channels at the same time [2]. Different from the ordinary internet connections, the WSNs is specialised in monitoring the environments and the energy of the nodes are limited by the battery capacity. Reducing the transmission delay and the energy consumption should be resolved with higher priority to raise up the system efficiency in supporting the continuous environment control and monitoring.

The WSNs present the characteristics of flexible connection infrastructure and the dynamic data transmission frequency because it is a self-organizable system. These characteristics result in the difficulty in the topology control in WSNs because of the topology changes are not easy to be predicted. Nevertheless, the topology control plays a key role in expanding the lifetime, reducing the interference and the packet retransmission of the WSNs [3]. Among all existing methods for the topology control, the tree-based topology structure is the most popular choice for gathering and aggregating the detected data [4–6]. In addition, many discussions on the virtual backbone structure composed of a Connected Dominating Set (CDS) are revealed, recently, because it presents high flexibility and verifiability [7,8]. Many existing literatures have proofed that the virtual backbone design can improve the routeing performance and reduce the energy consumption of the WSNs. However, the discussion on the packet collision and the transmission delay in the CDS based routeing algorithms still remains vacant. The packet collision occurs when two or more CHs sharing the same transmission channel attempt to deliver the packet to the same receiver node (also known as the father CH) at the same time. The

packet loss and the transmission delay may raise because of the collision and cause the negative impact of the WSNs performance.

In this paper, in order to minimize the delay and to avoid the collision, a CDS-based Data Aggregation Algorithm will be proposed. The core of this algorithm is to construct a *cross cube spanning tree* topology control algorithms for the WSNs. An  $n$ -cross Cube can be composed of  $N = 2^n$  nodes and it has a diameter equal to  $\lceil (n-1)/2 \rceil$  [13,29]. The network, which fits in the Cross Cube structure, has less diameter and higher fault tolerance capability than the ordinary hypercube structure with the same dimension. The main contribution of our work are concluded as follows:

1) A novel distributed induced tree of Crossed Cube based CDS Construction algorithm ( $C^4$ ) is proposed. The induced tree of Crossed Cube is constructed with the aims of decreasing the number of CDS. After theoretically analysing the crossed cube features, an optimal crossed cube spanning tree is revealed. The optimal crossed cube spanning tree is with the ability to eliminate the packet collision and minimising the transmission delay.

2) A collision-free data gathering protocol called the Cluster based Converge Cast Tree (CCCT) is proposed based on the Crossed Cube spanning tree features. The protocol performance is analysed theoretically and is followed by the simulations, which demonstrate that the CCCT protocol is efficient in avoiding packet collision and reducing the transmission delay. The simulation results indicate that the transmission delay can be reduced approximately 40 percent in average.

The rest of this paper is organized as follows: the related works in the topology control based data aggregation is reviewed in Section 2, to motivate the problem defined in Section 3. The proposed Cluster based Converge Cast Tree protocol is presented in Section 4. The theoretical analysis and the simulation results are discussed in Section 5 and Section 6, respectively. The paper is concluded in Section 7.

## 2 Related works

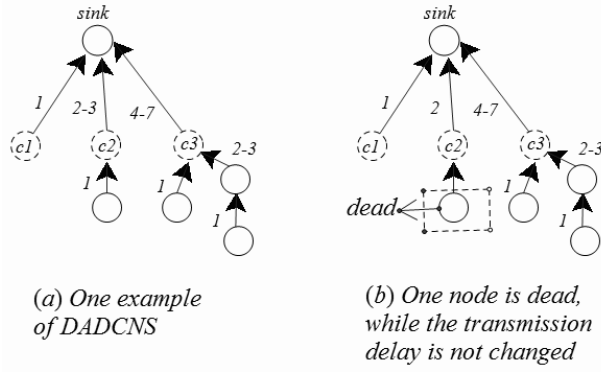
There are three important factors, including delay reduction, network lifetime prolongation, and transmission reliability, which are fundamental to the WSNs. Many efforts regarding these issues have been made in the related research fields [10,14,15]. For instance, Dong et al. [10] propose a data gathering protocol called the Broadcasting Combined with Multi-NACK/ACK (BCM-N/A) protocol, which presents the energy and the delay efficiency in the data gathering process in both the intra-cluster and the inter-cluster. Doudou et al. [16] propose a strategy based on placing two sink nodes and a few extra relay nodes, which are close to the sinks areas, in the WSNs. In their design, the data flow is alternatively sent to those two sinks in the sequential transmission cycles. However, both of the protocols mentioned above produce many extra transmission demands, which are not counted in the topology of the virtual backbone, for transmitting the NACK/ACK messages. On the other hand,

Charalampos et al. [14] present a data collection protocol for the WSNs by utilising a number of sink nodes with mobility. In order to achieve the most efficient way of delivering the sensory data from the terminal nodes to the mobile sink nodes, the sink nodes are limited to visit only an appropriate number of rendezvous nodes while the remaining nodes sending their data through multi-hop communication paths toward these sink nodes. Nevertheless, the transmission delay caused by the movement of sink nodes are not considered in this method. Velmani et al. [15] propose a Velocity Energy-efficient and Link-aware Cluster-Tree (VELCT) scheme for data collection in WSNs, which provides an efficient way for mitigating the problems of the coverage distance, the mobility, the transmission delay, the traffic, the tree intensity, and the end-to-end connection. Their proposed method, VELCT, constructs the Data Collection Tree (DCT) according to the cluster head's location. The data collection nodes in the DCT only equipped with the power source and the communication module. In other words, the data collection nodes are not capable of working as a sensor but simply are able to collect the data packets from the cluster heads and deliver them to the sink node.

On the other hand, using the clustering based algorithm to organise sensor nodes into clusters has attracted a lot of attention in recent days. The framework of the CDS is described as follows: the sensor nodes in the dominating set are called the Cluster Heads (CHs) or the dominators, and the connected nodes are responsible for forming the connections between the dominating sets after the CDS-based virtual backbone is constructed. The nodes, which are adjacent to a dominator, are called the Cluster Members (CMs) or the dominatees. The routing design can be simpler and is more capable of adapting to the topology changes because the data transmission is limited in the virtual backbone composed of the CHs. Within each cluster, one node is selected as the cluster head. The cluster head is responsible for 1) collecting data from its cluster members, 2) fusing the data by means of data/decision fusion techniques, and 3) reporting the fused data to the remote base station. In each cluster, the cluster head is the only node involved in the long distance communications. Thus, the total energy consumption of the whole network is therefore reduced.

The topology control algorithms focus on maximising the WSNs lifetime under the limited energy source [9,10] and minimising the number of nodes in the virtual backbone [8,11,12]. However, to the best of our knowledge, only a few of them utilise the CDS structure to drive a minimising delay topology control. The collision avoidance is yet discussed. Cheng et al. [17] propose a Delay-Aware Data Collection Network Structure (DADCNS), which aims at minimising the duration of a data collection process in the WSNs. A centralised control algorithm, which uses the top-down approach is proposed. The base station is assumed to have the coordinates of all sensor nodes in the network. The base station launches commands to the sensor nodes for establishing the essential data links and form the appropriate network structure. As shown in Fig. 1 (a), the number of nodes  $N$  in the proposed network structure has to be restricted to  $N = 2^k$ , where  $k = 1, 2, \dots$ . However, this structure is too regular

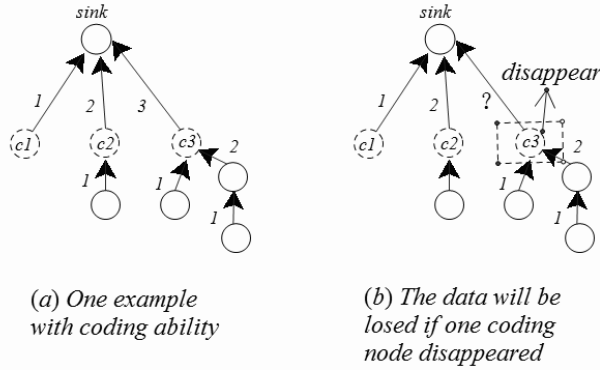
to maintain for the dynamic changing in the real-world scenario. For example, if the node framed with a rectangular, which is shown in Fig. 1 (b), is dead. The transmission delay is still 7 clocks for the sink node to receive data from the third Cluster Head  $C3$  without any update to a shorter delay because  $C3$  is previously designed to idle for 3 clocks to gain enough time for collecting the data from its cluster members. Without a proper update mechanism, the sink node will have more and more idle clocks when the terminal nodes start dying. The time for the idle clocks is simply wasted and becomes a burden on the system efficiency. Another drawback of this design is that the whole system is constructed under the assumption of all sensor node coordinates are known. However, the assumption is difficult to be satisfied in the real-world scenario.



**Fig. 1** Data collection in multi-hop network

Cheng et al. [18] propose another algorithm based on the network coding scheme. The sensor nodes with the irresponsibility of forwarding the received data are equipped with the coding ability. They encode all data it received into one message stream. Thus, the forwarded packages are merged into only one message as revealed in Fig. 2 (a). Hence, the number of clocks  $t(N)$  required for the sink node to collect data from the whole network is  $t(N) = \log_2 N$  and is equal to the number of clocks required by the cluster heads to collect data from every child node in their connection. However, the drawback of this scheme is that all transferred packets under the subnet of a cluster head may be completely lost once the cluster head is dead. The result of data loss in the subnet is not reversible. Fig. 2 (b) shows this scenario of data loss happened when the cluster head  $C3$  is out of function.

To reduce the data transmission distance, Khan et al. [19] introduce the mobile actor node, which is capable to physically move in the spatiotemporal space, to cruise to the region with the detected events. Since the data transmission distance is reduced from the terminal nodes to the mobile actor node is shrunk, the WSNs presents a better performance with the minimum delay and the high throughput. Nevertheless, the system has some premise that needs



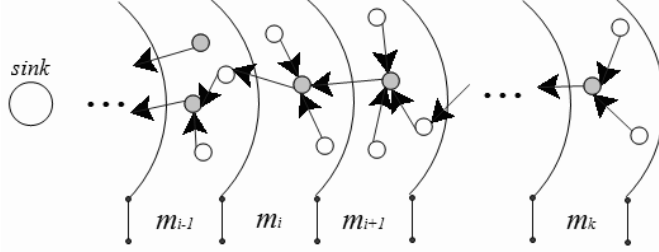
**Fig. 2** Data gathering by network coding

to be satisfied. For example, the WSNs covered area has to be divided into grids in the size of 20 times 20 meters. In addition, every grid is numbered with a unique label for identifying the sensor nodes' positions. Xu et al. [20] have revealed the theoretical proof and analysis regarding the delay of the aggregation schedule in Khan et al.s' method. Li et al. [21] propose a distributed scheduling method for generating the collision-free schedules, where  $D$  is the network diameter. However, the network diameter is too large to be applied in the real-world scenario. To our best knowledge, the cross cubes' network diameter is the smallest among the homogeneous network [13]. Sidera et al. [23] capture the tradeoff that exists in all mobile wireless delay-tolerant networks between the packet delivery delay and the aggregate transmission cost. Nevertheless, the consideration to manage the collision effect has remained vacant. Hong et al. [24] propose a hybrid beaconless geographic routeing (HBGR) protocol. In their method, the packets in the WSNs are split into the delay sensitive packet and the normal packet base on the needs of the applications. The HBGR protocol provides different mechanisms for dealing with different packets. However, the HBGR protocol heavily relies upon the geographic position information of the sensor nodes. The transmission overhead for the geographic information brings in extra power consumption and transmission requirements.

### 3 Performance Models

The main function of a sensor node is to report the detected event as the sensing data is periodically sent to the sink node within a valid time interval through multi-hop routing in the WSNs. The cluster based routing protocol is one of the efficient methods, where sensor nodes are divided into several groups and each group is called a cluster. One representative node is chosen in each cluster and the representative node is called cluster head (CH). The data is aggregated and forwarded at the CH so that it is finally delivered to the

sink. Fig. 3 shows the general system model for the clustering based routing in the WSNs. The data is forwarded to the sink node by the converge-cast tree structure. The cluster members transmit data to their CH. The CH in level  $m_{i+1}$  merges the received data by the network coding method before forwarding the collected data to its parent cluster head at level  $m_i$ .



**Fig. 3** The cluster based data converge-cast tree construction

### 3.1 The Communication Model and Assumptions

In this paper, we assume that the WSN has homogeneous communication model. Sensor nodes are deployed randomly and distributed according to a homogeneous spatial Poisson process with intensity  $\frac{N}{\pi r^2}$ , where  $N$  is the average number of neighbors and  $r$  is the sensor node radio range [22]. Since the link between any pair of nodes is bidirectional, the WSN is modeled as a connected bidirectional graph  $G = (V, E)$ , where  $V$  is a set of deployed sensor nodes and  $E$  is the set of edges.  $\forall u, v \in V$ , there exists an edge  $(u, v) \in E$  and  $(v, u) \in E$  if and only if  $u$  and  $v$  are mutually connected and there is no obstacle preventing radio wave transmission between  $u$  and  $v$ .

**Definition 1** (Node neighbourhoods) [8] Consider a node  $u \in V$ , the open set of 1-hop neighbor nodes covered by  $u$  is denoted by  $N_1(u)$ ,  $N_1(u) = \{v \mid (v, u) \in E\}$  and the close set of 1-hop neighbors of  $u$  is  $N[u] = N(u) \cup \{u\}$ . We also use  $N_2(u)$  to denote the 2-hop neighbors, i.e.,  $N_2(u) = N_1(u) \cup \bigcup_{v \in N_1(u)} N_1(v)$

**Definition 2** (Maximal Independent Set (MIS)) [8] A Maximal Independent Set of a graph  $G = (V, E)$  is a subset  $V' \subseteq V(G)$  such that every pair of vertices in  $V'$  are not adjacent, and no independent vertex can be added into  $V'$ .

**Definition 3** (Dominating Set (DS)) [7] A Dominating Set of a graph  $G = (V, E)$  is a set of nodes  $V^d \subseteq V(G)$  such that for every  $(u, v) \in E(G)$ ,  $u \in V^d$  or  $v \in V^d$ .

**Definition 4** (Connected Dominating Set (CDS)) [8] A Connected Dominating Set of a graph  $G = (V, E)$  is a DS of  $G$  such that the subgraph of  $G$  induced by the nodes in this set is connected.



In many cases, an MIS construction algorithm is used for finding a DS. The nodes in CDS are called the dominators; on the other hand, nodes, which are not the dominators are called the dominatees. The size of CDS equals to the number of the dominators. Our proposed algorithm is utilized to find the MIS nodes by building an Inducted Tree of the Crossed Cube network. The found MIS nodes are further connected together to construct a CDS.

**Definition 5** (Network Coding (NC)) [27] At each network coding node  $v$ , the received messages  $x_{u_1}, \dots, x_{u_k}$  are firstly combined by a coding method

$$f_v(x_{u_1}, \dots, x_{u_k})$$

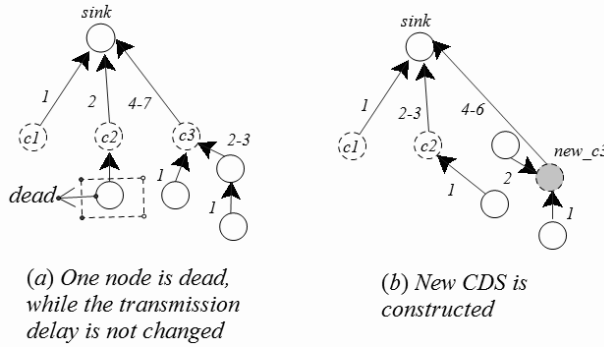
and the combined message is forwarded towards its destinations after the coding process.

Since the coding nodes combine the data flows by the mathematical coding operations before forwarding the encoded packets rather than sending a large number of originally collected small packets. It reduces the overheads for the communication and is able to remarkably reduce the energy consumption. [28]

### 3.2 Problem Definition

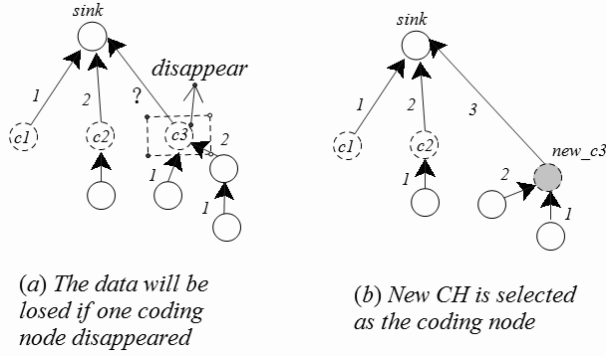
In this work, the main problems are: 1) in CDS and tree based structure WSN, give the transport delay under reliability constraint from theoretical analysis; 2) How to further decrease the network delay and improve the transport delay when some error happened in the CDS and tree based structure. That is to say, a fault tolerance structure should be constructed. Since we aim to minimize delay in WSNs, the definition of the delay is given as follows.

**Definition 6** (Transport delay) [10] The transport delay is defined as the time from a packets first transmission until its successful arrival at the sink.



**Fig. 4** The method for some nodes dead

We also use Fig. 1 and Fig. 2 as examples. In order to improve the transport delay, when error happened, new CDS should be constructed, which is fault tolerance. As shown in Fig. 4, when the CM of C2 disappeared, one of the CM of C3 closest to C2 will become the CM, then the C3 can forward the data in slot 4-6. As shown in Fig. 5, when the CH C3 disappeared, its CMs' data can not be forwarded to the sink, then a new CH *new\_C3* will be selected to forward these data.



**Fig. 5** The method for some CHs disappeared

#### 4 CDS-based Data Aggregation Algorithm

Most beaconless cluster routing protocols use traditional handshaking to select forwarder [24]. Once a sensor has sensing data and need to send the message, it begins carrier sense until channel is clear for a certain time. It is quit unsuitable for large-scale WSNs. By contrast, CDS can divides a wireless sensor network into several clusters. Within each cluster, the dominating node is as a Cluster Head (CH) and the rest are as Cluster Members (CM). The cluster head will collect data from its cluster members directly or in a multi-hop manner. In this section, a CDS-based Converge Cast Tree (CCCT) protocol will be proposed. In order to understand CCCT better, the overview of the protocol is given at first, then we will describe each part of the algorithm in detail. There are four steps in this protocol:

1) CDS construction: CDS has become a well known approach for constructing a virtual backbone in wireless sensor networks. Traffic can forwarded by virtual backbone and other nodes turn off their radios to save energy. However, the size of the CDS can not be too large. Since there are many packets transmit in the virtual backbone, the interference will be incurred in the CDS. We need to construct the CDS as small as possible. As we know, constructing a minimum CDS is an NP-hard problem, and thus most researchers concentrate on how to construct approximate algorithms. In this section, Maximal

Independent Set (MIS) will be used to form a approximate optimal CDS. The priority of a node who will be selected into CDS is determined according to a new parameter, namely the degree of the node in the square of a graph, which is consistent with the concept of MIS.

2) Inter-cluster transportation: The nodes in the network are divided into multiple clusters, each comprising a CH and cluster members that communicate to the CH. Data of CH is sent to the sink via multi-hop among CHs.

3) Intra-cluster transportation: The top-down approach is a kind of centralized control algorithm. The CH in each cluster can construct a rank level in each cluster. At the end of the optimization process, the CH will instruct the sensor nodes to establish the essential data links and form the appropriate network structure for its own cluster. Coding nodes mix data flow using mathematical operations, and then forward only encoded packets instead of a large number of original information.

4) CDS reconstruction: For a fault tolerance structure, new CHs should be selected when there are some errors happen in some of the clusters. It is desired that new CDS should be constructed using the available local information such that the total coverage of the network by CDS is higher than the coverage threshold  $\eta$ , which can be set adaptively.

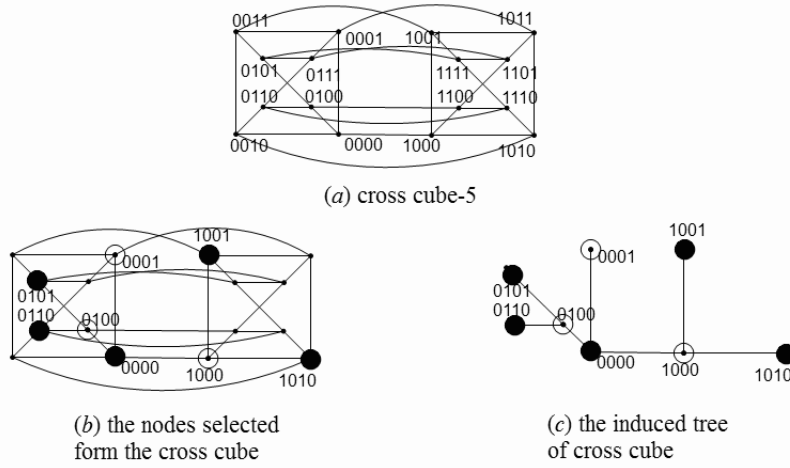
#### 4.1 Cross Cube based CDS Constructed algorithm

This section describes the CDS construction algorithm in detail. There are three stages in the Cross Cube base CDS Constructed ( $C^4$ ) algorithm. The first stage is to construct the square of the graph, according to the concept of MIS; The second stage is to find the approximate optimal MIS based on  $C^4$  algorithm, which is constructed in local square graph, and then to form a CDS by connecting the nodes in MIS. The third stage is to construct the spanning tree for the cross cube in order to avoid the collision.

Base on the concept Induced Tree of Crossed Cube, the  $C^4$  algorithm which was proposed in our previous article [8]. Fig. 6 shows the example of  $CQ_5$  (Fig. 6(a)) and the induced tree of cross cube  $T_5$  (Fig. 6(b) and (c)).

At the initial stage, Each node has to be given an initial status and then exchange neighbour information among themselves to get ready for the CDS construction. Each node should construct the local graphs  $G$ ,  $G^2$  and  $G^*$  with itself as the center. Each node  $u_i$  holds the message  $(id, W(u_i))$ , where  $W(u_i) = (f(u_i), d_{G^*}(u_i), d_G(u_i))$  is the weight of the corresponding node, and  $f(u_i)$  is the identification of the node status, the numerical value can be obtained by formula (1).

$$f(u_i) = \begin{cases} 0, & \text{if } u_i \text{ is a normal node} \\ 1, & \text{if } u_i \text{ is in the DS} \\ 2, & \text{if } u_i \text{ is a connected node} \\ 3, & \text{if } u_i \text{ has been eliminated} \end{cases} \quad (1)$$



**Fig. 6** Cross Cube-5 and its induced tree of cross cube

The degree of the node  $u_i$  in graph  $G$  and its square graph  $G^*$  are denoted as  $d_G(u_i)$  and  $d_{G^*}(u_i)$ , respectively. The node's weight is the priority for these elements, for example, some nodes with the same value of  $f$ , the higher the degree of  $d_{G^*}$ , the greater the weight. In other words, the probability of the node to be elected into the CDS is greater. Detailed description will be given in Algorithm 1, Cross Cube base CDS Constructed  $C^4$  algorithm.

Initialization for the parameters should be conducted before the algorithm starts, after that, one node is selected to join into the graph of Induced Trees of the Crossed Cube  $G_T$ , assume node  $u_0$  has the maximum weight, which labeled by '0'. A mask named as *flag* is used to describe whether the algorithm is completed, it is also implemented by judge whether all the nodes in the network have decided their own states, as a dominator or a dominee. When the algorithm is completed, *flag* is set as value 1. Otherwise, the 1-dimension Induced Tree of the Crossed Cube  $T_1$  should be constructed. One sensor node  $u_1$ , which has the maximum weight, is selected from  $N_{G^*}(u_0)$ , labeled by value 1, and set  $f(u_1) \leftarrow 1$ , this new node  $u_1$  and the related edge are added into Induced Tree  $V(T_1)$  at the same time. Then, the new Induced Tree has constructed as  $V(T_n) \leftarrow \{u_0, u_1\}$  and  $E(T_n) \leftarrow E(T_n) \cup (u_0, u_1)$ . According to the definition, if induced tree in our previous work [8], the dimension of the Induced Tree can be extended. The nodes should be divided into bipartite partitions, let  $V_1(T_n) \leftarrow \{u_1\}$  and  $V_2(T_n) \leftarrow \{u_0\}$ .

If it is still  $flag = 1$ , the 2-dimension Induced Tree should be constructed, which is similar to construct the 1-dimension Induced Tree. After that, the nodes and edges are selected, where  $V(T_n) \leftarrow \{u_0, u_1, u_2\}$  and  $E(T_n) \leftarrow E(T_n) \cup (u_0, u_2)$ , these nodes will be divided into bipartite partitions as  $V_1(T_n) \leftarrow \{u_0\}$  and  $V_2(T_n) \leftarrow \{u_1, u_2\}$ . If it is still  $flag = 1$ , the  $n$ -dimension Induced Tree can be extended.

**Algorithm 1:**  $C^4$  algorithm

---

```

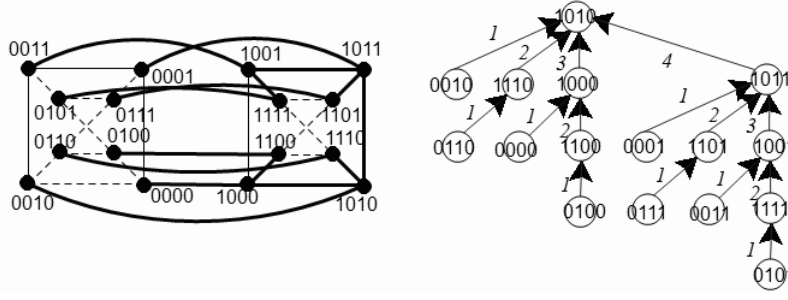
1  $V(T_n) = \phi; E(T_n) = \phi; CDS = \phi$ , the set of the dominatees  $V_d = \phi$ ;  $flag = 0$ ;  $n = 1$ 
2 for  $u_i \in V$  do
3   | calculate its weight  $W(u_i) = (f(u_i), d_{G^*}(u_i), d_G(u_i), re(u_i))$ , set  $f(u_i) = 0$ ;
4 end
5 select a node  $u_0$  randomly and labeled by '0', set  $f(u_0) = 1$ ,  $V(T_n) = u_0$ ;
6 if ( $flag \neq 1$ ) construction 1-dimensional induced tree; and add all selected node
   into  $V(T_n)$ , then consider whether all node is dominated by  $V(T_n)$ , if so  $flag = 1$ ;
   then
7 end
8 if ( $flag \neq 1$ )  $n = 2$ ; construction 2-dimensional induced tree; and add all selected
   node into  $V(T_n)$ , consider whether all node is dominated by  $V(T_n)$ , if so  $flag = 1$ ;
   then
9 end
10 while ( $flag \neq 1$ ) do
11    $n++$ ;
12   for  $u_i \in V_1(T_{n-2}) \notin \phi$  or  $v_i \in V_2(T_{n-2}) \notin \phi$  and  $flag \neq 1$  and
      $N_{G^*}(u_i) \setminus (V_d \cup CDS) \neq \phi$  and  $N_{G^*}(v_i) \setminus (V_d \cup CDS) \neq \phi$  do
13     |  $i = 0$ ;
14     |  $V(T_n) = V(T_{n-2})$ ;
15     | select a node in  $N_{G^*}(u_i) \setminus (V_d \cup CDS)$ , with the maximum weight  $u_j$  with
        $f(u_j) = 1$ ;
16     | select a node in  $N_{G^*}(v_i) \setminus (V_d \cup CDS)$ , with the maximum weight  $v_j$  with
        $f(v_j) = 1$ ;
17     |  $V(T_n) = V(T_n) \cup u_j$ ,  $E(T_n) = E(T_n) \cup (u_i, u_j)$ ;
18     |  $V(T_n) = V(T_n) \cup v_j$ ,  $E(T_n) = E(T_n) \cup (v_i, v_j)$ ;
19     |  $i++$ ;
20     | call function connect( $V(T_n)$ ); consider whether all node is dominated by
        $CDS$ , if so  $flag = 1$ ;
21   end
22   for  $u_i \in V_1(T_{n-2})$  or  $v_i \in V_2(T_{n-2})$  or do
23     |  $u_i$  labeled by  $u_i00$ ;  $v_i$  labeled by  $v_i00$ 
24   end
25   for  $u_j$  code of  $u_j01$  do
26     |  $v_j$  labeled by  $v_j01$ , and  $v_k$  labeled by  $v_k10$ ;
27   end
28    $V_1(T_n) = \{u_j01, v_i00 : u_j \in V_1(T_{n-2}), v_i \in V_2(T_{n-2})\}$ ,
      $V_2(T_n) = V(T_n) - V_1(T_n)$ ;
29 end

```

---

When at least two nodes are selected into the Induced Tree as DS, a function connect ( $G_T$ ) is called, which aims to select the connector nodes to make the DS connected. This function implementations as follows. For each edge  $(u, v) \in E(G^*)$ , according to the definition of graph  $G^*$ , there exist some nodes in  $N_G(u) \cap N_G(v)$  if  $f(u) = f(v) = 1$ , the one who has the maximum weight named  $w$  can be selected as the connector node, thus, node  $w$  joins into the set of CDS referred to as  $CDS \leftarrow V(T_n) \cup w$ , and we set  $f(w) \leftarrow 2$ ,  $C_1 \leftarrow CDS$  at the same time. Later, the algorithm also should be judged whether is completed.

Next, spanning trees of crossed cube will be constructed. The nodes which is in the induced tree but not in the spanning tree, can be selected following our



**Fig. 7**  $C^4$  based converge-cave tree construction

previous works [30]. Then the spanning tree can be constructed as following: At first, one CH is selected as the root of the spanning tree, next, we change one or two bits of the label in turn. For example, if the CH's label is '1010', its sons' labels are '0010', '1110', '1000' and '1011'. If these nodes are not exist in the induced tree, they can be elected from the CMs of the CH.

#### 4.2 Inter-cluster Transportation

The data collection tree inter-cluster is an acyclic graph, where sink node is the root of  $T$  at level 0. In a rooted spanning tree  $T$ , nodes  $v_i$  and  $v_j$  are siblings if they share the same parent. Assume there are  $|N_{ch}|$  cluster members in a cluster, then a tree with the high  $\lceil \log_2 |N_{ch}| \rceil$  is builded, which is used for data collection.

The nodes in the network are divided into multiple clusters, each comprising a CH and cluster members that communicate via one hop to the CH. Data of CH is sent to the sink via multi-hop among CHs. A CH with level  $p$  will form  $p-1$  data links with  $p-1$  CHs, which these CHs are with different levels starting from 1, 2,  $\dots$  up to  $p-1$ . All these  $p-1$  CHs may become the child nodes of the node with level  $p$ . The node with level  $k$  will form a data link with a node with a higher level. This higher level CH will become the parent CH of the node with level  $k$ . The sink will be considered as a special case, it is the one with the highest level in the network.

Based on spanning tree in cross cube, the message forward rule should be set. Since the CDS constructed algorithm is builded by Spanning Trees of the Crossed Cube network, then the CDS is a tree based construction and each cluster head has its own label, and which is unique. Since the tree construction make sure there is no situation of the link intersection, and each cluster head can forward its message to its parents cluster head by some rules. Since each CH has its own label, according to the routing to the sink builded by the  $C^4$  algorithm, then we define that the CH who has the lower label sent the message first, when conflict happened in one channel.

An example of the proposed tree with  $|N_{cm}| = 16$  is shown in Fig. 7. In this example, it take  $4 \times T$  for the base station to collect all data from 16 nodes.

The leaf nodes forward their data to their father nodes at the first slot time. For example, at slot one, node with make number '0110' forwards data to its father node '1110', node with make number '0100' forwards data to its father node '1100'. At slot 2, node with make number '1100' forwards data to its father node '1000', node with make number '1101' forwards data to its father node '1011'. It is easy to see that there is no collision happened in this kind of construction. The inter-cluster transportation is based on the construction algorithm of spanning tree shown in Algorithm 2.

Although there are many spanning trees in one cross cube, it will be decided as long as the root is selected.

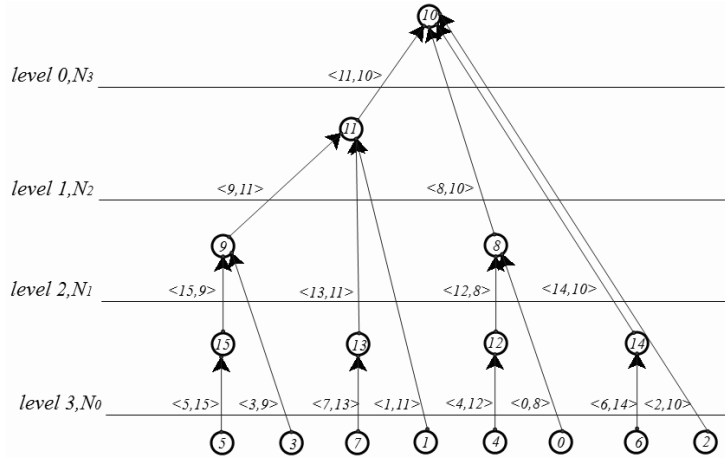


Fig. 8 The inter-cluster transportation by the tree construction

---

**Algorithm 2:** Spanning tree based Crossed Cube construction

---

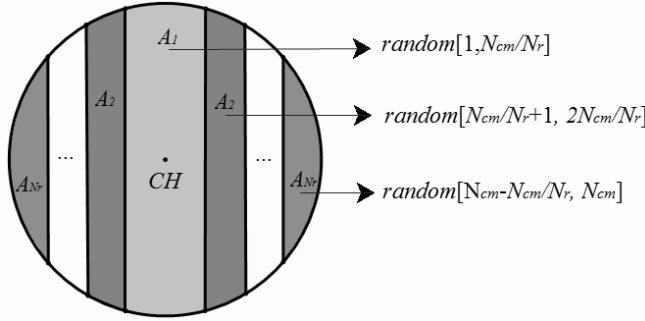
- 1 choice one  $CH_0$  as the root of the spanning tree of the cross cube;  $V(T) = CH_0$ ;  
 $E(T) = \phi$ ;  $S = \log_2 |N_{cm}|$ ;
  - 2 each CHs decide their levels according to the spanning tree of cross cube.
  - 3 **for**  $level = 0$  to  $S - 1$  **do**
  - 4     **for**  $x \in V(T)$  **do**
  - 5         **end**
  - 6          $y = N_{S-level-1}(x)$ ;
  - 7          $E(T) = E(T) \cup \langle y, x \rangle$ ;
  - 8          $V(T) = V(T) \cup y$ ;
  - 9 **end**
- 

For example, in Fig. 8, suppose that vertices are labeled with binary notation. There is only one vertex '10' in  $V(T)$ , during the first iteration ( $level = 0$ ), vertex  $N_3('10') = '11'$ . As a result,  $V(T) = \{'10', '11'\}$  and

$E(T) = \langle '11', '10' \rangle$ . During the second iteration (level=1), each vertex  $v$  in  $V(T)$  is connected to vertex  $N_2(v)$ , the iteration will be stopped until each cluster-head is connected when  $level = S - 1$ .

#### 4.3 Intra-cluster transportation

After the CDS is constructed, and each node determines its state, as a CH or CM. Each cluster head knows the number of its cluster members. Assume there are  $|N_{cm}|$  cluster members in a cluster. Those CM is used for data collection intra-cluster. The CH area is divided into  $N_r$  regions  $A_1, A_2, \dots, A_{N_r}$  based on the lines, such that all CM in  $A_i$  will be given a rank, which is an integer between 1 and  $N_{cm}$ . CMs are with different ranks starting from 1, 2, ... up to  $N_{cm}$ . All these  $N_{cm}$  sensors will become the child nodes of CH.



**Fig. 9** The intra-cluster transportation

Fig. 9 illustrates the rank dividing mechanism. CMs in regions  $A_1$  random select the rank between 1 and  $\frac{N_{cm}}{N_r}$  as initial, CMs in regions  $A_2$  select the rank between  $\frac{N_{cm}}{N_r} + 1$  and  $2\frac{N_{cm}}{N_r}$  as initial. The rank decides the CMs' time slot, that is when they can forward the data to their CHs. The higher the rank is, the more earlier it sends the data. By this way, the collision can be avoided.

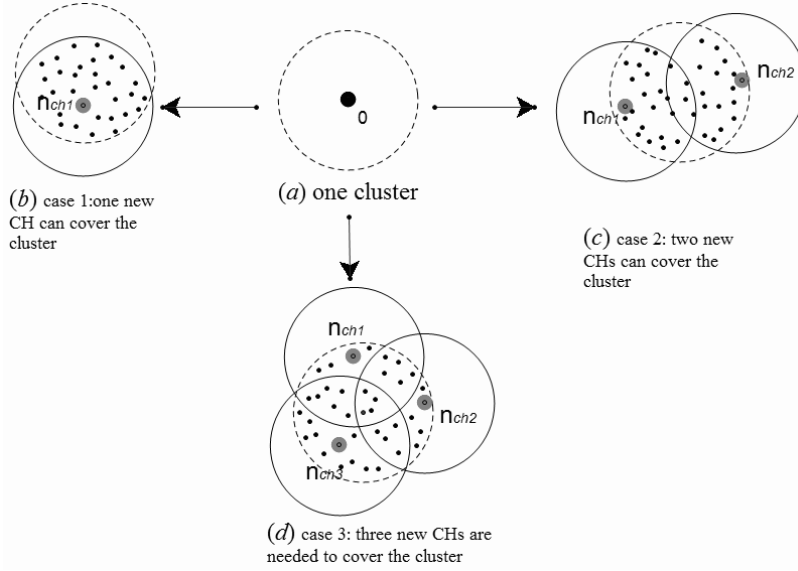
Furthermore, since coding nodes mix data flow using mathematical operations, and then forward only encoded packets instead of a large number of the original information, it can reduce energy consumption remarkably. Network coding becomes widely significant after it was first proposed by Ahlswede et al. [25]. Many researchers consider it as an efficient technology for improving network performance [26]. The number of message that be encoded by coding node in each transmission is called as encoding number. Maximize the encoding number is most researches. After the CDS construction, the CHs are act as the coding nodes. They can coding all the received message, then forward them to the father CH. In this paper, we do not consider the network coding algorithm in detail, we use the basis linear network coding algorithm.



#### 4.4 CDS reconstruction

Two kinds of mistakes will be happened in one cluster. One is CH error, the other is the CM disappeared. If the CM is disappeared, then the CH find there is no member in its cluster, it will broadcast the member recruit message. The closed node who satisfies the condition as CM, it will sent the message for asking join into the new cluster. If the error is happened in CH, then new cluster head should be elected. It is desired that new CDS should be constructed using the available local information such that the total coverage of the network by CDS is higher than the threshold  $\eta$ .

For  $i > 1$ , the new CDS is constructed by selecting cluster-head in each cluster. One cluster example is shown in Fig. 10 (a), node  $o$  is the cluster-head which is selected as a node in the CDS by  $C^4$  algorithm. When node  $o$  has not enough energy as a cluster-head, it must be replaced. Then the new cluster heads may be constructed. Same cases could happen in each cluster. For instance, three cases are illustrated in Figs. 10 (b) - (d).



**Fig. 10** An example of reconstructing clusters

Case 1. As shown in Fig. 10 (b), only one node is needed. If node  $a_1$  is selected as a cluster-head, which can cover all the nodes that node  $o$  covered. Sub-Induced Tree does not need to be constructed.

Case 2. Two non-adjacent nodes can be selected as the cluster-heads, such as nodes  $b_1$  and  $b_2$  which are shown in Fig. 10 (c). 1-dimensional sub-Induced Tree needs to be constructed in this cluster.

Case 3. If two nodes still cannot cover those nodes, three non-adjacent nodes can be selected as the cluster-heads, such as nodes  $c_1$ ,  $c_2$  and  $c_3$  illustrated in Fig. 10 (d).

In fact, at most five nodes need to be selected to cover all of the nodes in the same cluster. In order to guarantee the coverage intensity, 2- or 3-dimensional sub-Induced Trees must be constructed in these clusters.

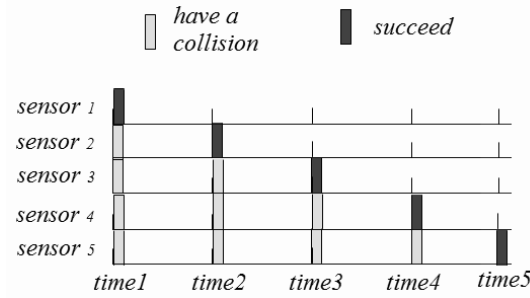
## 5 Theoretical analysis

Under the CCCT protocol, after each data gathering in intra-cluster and the converge-cast by CHs to the sink, such process is called one round. The following theorem and simulations give the data load of node in one round. Theorem 1 proves the correctness of the algorithm, and Theorem 2 proves that our CCCT protocol can avoid the collision efficiently. Lemma 1 and Theorem 3 prove the delay time slots approximation ratio complete to the MIS; Lemma 2 and Theorem 4 give the value of the delay which relate to the number of the node and the dimensionality of the cross cube.

**Theorem 1** *The CDS constructed by algorithm  $C^4$  is a spanning tree based cross cube and root at  $u$  with the height  $n + 1$  in  $CQ_n$ .*

*Proof* By algorithm  $C^4$ , it is easy to verify that there are  $1 + 2^0 + 2^1 + \dots + 2^{n-1} = 2^n$  vertices in the tree. Since  $V(CQ_n) = 2^n$  and the initial vertex  $N_i(u)$  in the tree have  $n$  different neighbours including a leaf vertex  $u$ , the the height of the tree rooted at  $u$  is  $n + 1$ . Thus, the theorem holds. ■

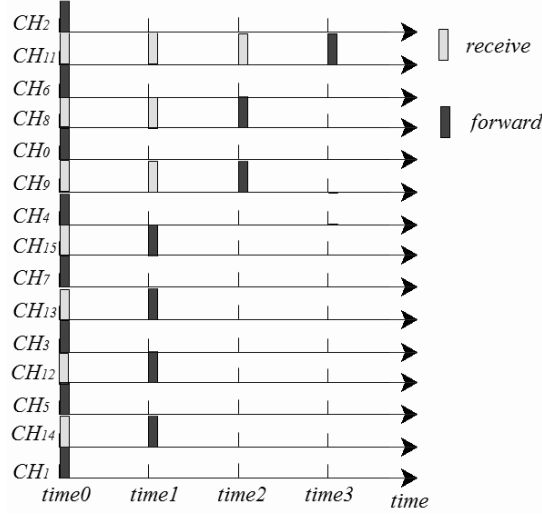
**Theorem 2** *Under the CCCT protocol and in the CH-based virtual backbone network, there is no collision happen.*



**Fig. 11** Some collision

*Proof* Fig. 11 shows the protocol proposed by [16], which is the communication system with many collisions, it requires application delay 5 clock cycles for 5 sensors. With the increase of the number of the sensors, the collisions will be

much higher. However, as shown in Fig. 12, in a spanning tree of cross cube, every cluster heads receive and forward messages in different clock cycles. For example, at clock cycle 1, CHs 1, 3, 5, 7, 9, 11, 13, 15 forwards message to their own parent CHs. For CH 2, it receives messages from CH 3 at clock cycle 1, receives messages from CH 10 at clock cycle 2, receives messages from CH 6 at clock cycle 3 and forwards the coded messages to the sink in clock cycle 4. There is no collision happened in this spanning tree.



**Fig. 12** No collision happen in the spanning tree

Consider a cluster, given a period of  $t$  time slots, a CH can collect data from at most  $t$  directly connected child nodes, as long as these child nodes use different time slots to communicate. Within these  $t$  child nodes, the  $u^{th}$  node will report data at time slot  $u$ , which implies the  $u^{th}$  node can collect data from at most  $u - 1$  directly connected child nodes of itself before it has to report data to its parent node. Therefore, for a period of  $t$  time slots, a CH can receive data from at most  $2^t$  nodes (including itself). On the other hand, the minimum number of time slots required for a parent node to collect data from nodes (including itself) is  $\log_2 N$ . Thus, the minimum number of time slots required for a sink to collect data from nodes  $N$  is  $t(N) \leq \log_2 N$ . ■

Later, we will analysis the value of the delay in detail.

**Lemma 1** [8] *Given a graph  $G = (V, E)$ , let  $\alpha$  be the DS obtained from the  $C^4$  algorithm. Then*

$$\alpha \leq 3.2833\gamma + 4.5590 \quad (2)$$

where  $\gamma$  is the size of any optimal CDS of  $G$ .

**Theorem 3** *Given a graph  $G = (V, E)$ , the upper bound of time slots required for a base station to collect data from nodes  $N$  is*

$$t(N) \leq \log_2 \frac{N}{(3.2833\gamma + 4.559)} + \log_2 8.64\gamma \quad (3)$$

where  $N \geq 1$  is the number of the nodes in WSN, and  $\gamma$  is the MIS.

*Proof* According to Theorem 1, the minimum number of time slots required for a base station to collect data from nodes  $N$  is  $t(N) \leq \log_2 N$ .

In the CDS-based structure, each cluster has average  $\frac{N}{|DS|}$  sensors, then the total minimum number of time slots is:

$$t(N) \leq \log_2 N / |DS| + |CDS|. \quad (4)$$

According to Lemma 1, the Dominating Set has the size at most

$$|D_1| \leq 3.2833\gamma + 4.5590. \quad (5)$$

then the total minimum number of time slots is:

$$t(N) \leq \log_2 \frac{N}{(3.2833\gamma + 4.559)} + \log_2 8.64\gamma \quad (6)$$

where  $N \geq 1$  is the number of nodes in WSN, and  $\gamma$  is the MIS. ■

Though Theorem 1 proves the upper bound of delay time slots' approximation ratio complete to the MIS. However it is not intuitionally. Lemma 2 and Theorem 1 give the value of the delay which is related to the number of the node and the dimensionality of the cross cube.

**Lemma 2** [8] *Given a graph  $G = (V, E)$ , the number of nodes in the DS based on the Induced Tree of the Crossed Cube is approximately equal to*

$$0.447 \times [1.618^{n+2} - (-0.618)^{n+2}] \quad (7)$$

where  $n \geq 1$  is the dimensional of the Crossed Cube.

**Theorem 4** *Given a graph  $G = (V, E)$ , the maximum number of time slots required for a base station to collect data from node  $N$  is*

$$\frac{\log_2 N}{0.447 \times (1.618^{n+2} - (-0.618)^{n+2})} + n \quad (8)$$

where  $n \geq 1$  is the dimensional of the Crossed Cube.

*Proof* According to Lemma 1, the Dominating Set has the size at most

$$|D_1| \leq 3.2833\gamma + 4.5590. \quad (9)$$

According to Lemma 2, the number of DS constructed by our  $C^4$  algorithm similar to the following inequality

$$|D_1| \approx 0.447 \times [1.618^{n+2} - (-0.618)^{n+2}]. \quad (10)$$

The upper bound of dimensional  $n$  can be calculated by formulas (9) and (10), thus there is

$$\begin{aligned} 0.447 \times [1.618^{n+2} - (-0.618)^{n+2}] &\leq 3.2833\gamma + 4.5590 \\ \Leftrightarrow 2.618 \times 1.618^n - 0.382(-0.618)^n &\leq 7.35\gamma + 10.2, \end{aligned} \quad (11)$$

and then

$$1.618^n - 0.156(-0.618)^n \leq 2.9\gamma + 3.9. \quad (12)$$

Since

$$-0.06 \leq -0.156(-0.618)^n \leq 0.1, \quad (13)$$

and

$$n \leq \log_{1.618}(2.9\gamma + 3.96), \quad (14)$$

and then  $n \leq \log_2(4.64\gamma + 4)$ . It can be simplified as inequality (15) with  $\gamma \geq 1$

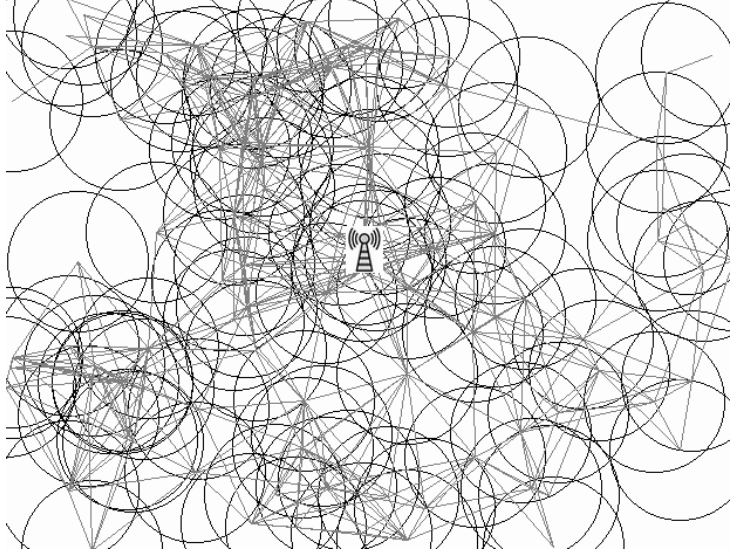
$$n \leq \log_2(8.64\gamma). \quad (15)$$

According to the Theorem 1, the minimum number of time slots required for a base station to collect data from nodes  $N$  is

$$\frac{\log_2 N}{0.447 \times (1.618^{n+2} - (-0.618)^{n+2})} + n \quad (16)$$

where  $n \geq 1$  is the dimensional of the Crossed Cube. ■

As far as we known, the cross cubes' network diameter is the smallest one among the homogeneous network. Next section shows that the simulations corresponds with the theoretical value.



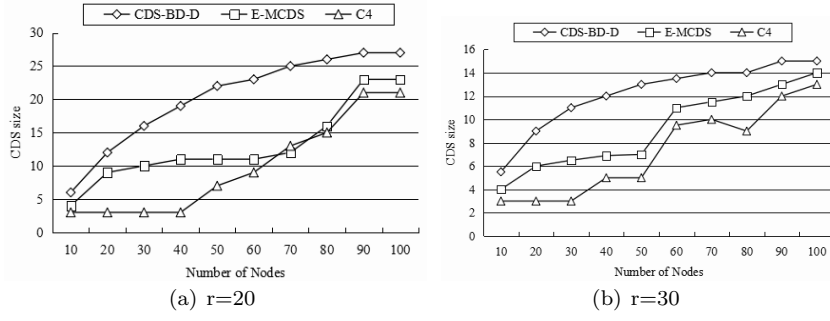
**Fig. 13** The data transmission in CDS based networks

## 6 Simulations

To verify the idea on the scheduling data transmission, the proposed algorithms are deployed in a WSN environment shown in Fig. 13. It contains  $n$  homogeneous sensor nodes, which are distributed in a region with a sink node located at the centre of the whole system. The deployment of the sensor nodes follows a homogeneous Poisson point process with a density of  $\rho$  nodes per unit area. The sensor nodes in the network are divided into several clusters. The CHs transmit data back to the sink node hop-by-hop in a designed sequence. The transmitting radius of a sensor node is denoted by the symbol  $R$ . For every node  $n_i$ , the probability that data transmission from node  $n_i$  to node  $n_j$  is denoted by the symbol  $p_i$ . The transmission is defined to take 1 clock cycle to complete the transmission of a single packet.

### 6.1 Competition of the size of the CDS

To evaluate the performance of the designed algorithm with different CDS sizes, the proposed  $C^4$  algorithm is compared with the most recently published algorithms, e.g. the CDS-BD-D [11] and E-MCDS [12]. The sensor node distribution presented in the figure is averaged over 100 simulations. The number of sensor nodes deployed in the space and the capable transmission range of the sensor nodes is incrementally varied from 10 to 100 by 10 and from 20 to 30, respectively, in each simulation. Fig. 14 shows the comparison of the algorithms with different CDS size. An observable phenomenon in Fig. 14 is that the required CDS size of  $C^4$  algorithm is the smallest among all compared



**Fig. 14** The comparison of the algorithms in terms of CDS size

algorithms. In addition, the size of CDS grows as the number of sensor nodes in the system increases. By comparing Fig. 14 (a) and Fig. 14 (b), the trend of the CDS size decreases while the transmission range of the sensor node increases. The reason is that a sensor node equipped with higher transmission range provides a larger coverage area. Since the whole WSN coverage area is finite, the CDS size can be reduced, accordingly. Based on the results given in the figures, which are obtained from the simulations, we conclude that the proposed  $C^4$  algorithm is capable of providing the same system coverage and data transmission ability with the smallest CDS size.

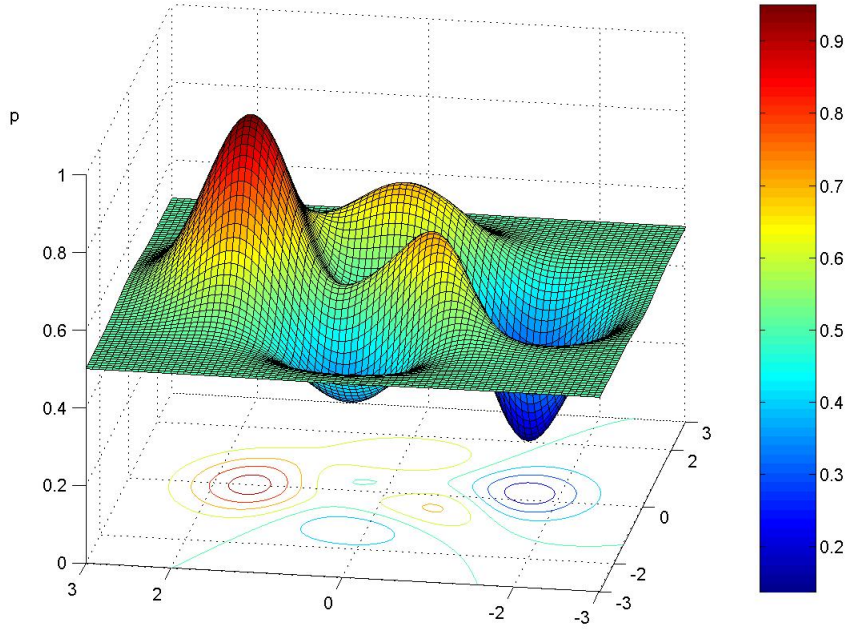
Moreover, several tests on the data transmission delay with different algorithms are also carried out in the simulations. Again, our proposed algorithm is compared with the CDS-BD-D and the E-MCDS based data collection algorithms.

The probability  $p_i$  of every node is calculated in a sensing and reporting cycle. In order to simulate the reality in the real-world scenario, we assume that the probability  $p_i$  fits in a multi-peak Gaussian Distribution, which can be described by Formula (17). The simulation results are presented with the average values. In addition, the designated area in the simulation environment is shown in Fig. 15.

$$\begin{aligned}
 P(x, y) = & \frac{(1-x)^2}{4} e^{-x^2-(y+1)^2} \\
 & - \frac{5}{6} \left( \frac{x}{5} - x^3 - y^5 \right) e^{-x^2-y^2} \\
 & - \frac{1}{36} e^{-(x+1)^2-y^2} + \frac{1}{2}.
 \end{aligned} \tag{17}$$

Events of the packet transmission/receiving and handling the collision all require clock cycles (the units for calculating the process time) for the WSN's wireless communication system to execute the coordinating procedures. The clock cycles required for transmitting the packet is calculated by Formula (18).

$$T = T_s \times \alpha_1 + T_c \times \alpha_2. \tag{18}$$



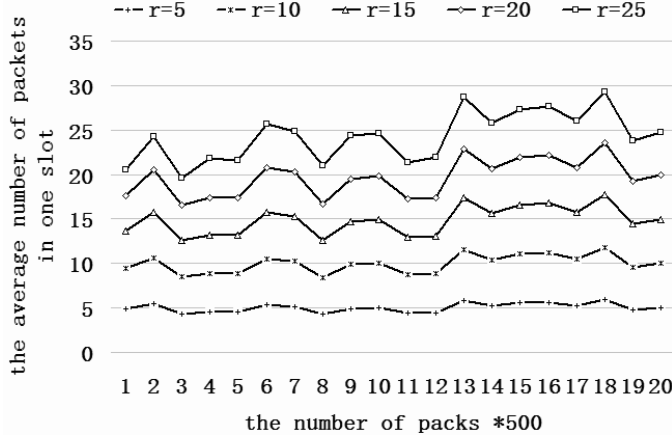
**Fig. 15** The probability that data transmission  $p_i$

where  $T_s$  denotes the time for transporting a package, and  $T_c$  denotes the time occur by collision.  $\alpha_1$  and  $\alpha_2$  denote the weight coefficients. In our simulation, the delay slot  $T_c$  is set as 20 percents of the transport slots  $T_s$ , and the weigher coefficients are set as  $\alpha_1 = \alpha_2 = 0.5$ .

## 6.2 Throughput of the CCCT algorithm

The task processing ability of the WSN is called the throughput, which is defined as the total number of bits processed per second. The throughput of a WSN containing 100 sensor nodes is shown in Fig. 16. The transmission radius of the sensor nodes is varied from  $r = 5$  to  $r = 25$  by increasing 5 units, respectively. The occupied number of clock cycles are averaged over 50 repeats. In every repeat, the WSN is randomly assigned to handle 500 – 10000 packets from the terminal nodes to the sink node. According to the simulation results, the throughput of WSN with different algorithms is almost identical because the number of cluster members in the CDS, of which the size is decided by the transmission radius, is almost the same. As the matter of fact, more packets can be collected by the CHs when the throughput is increased by the increment of the transmission radius.



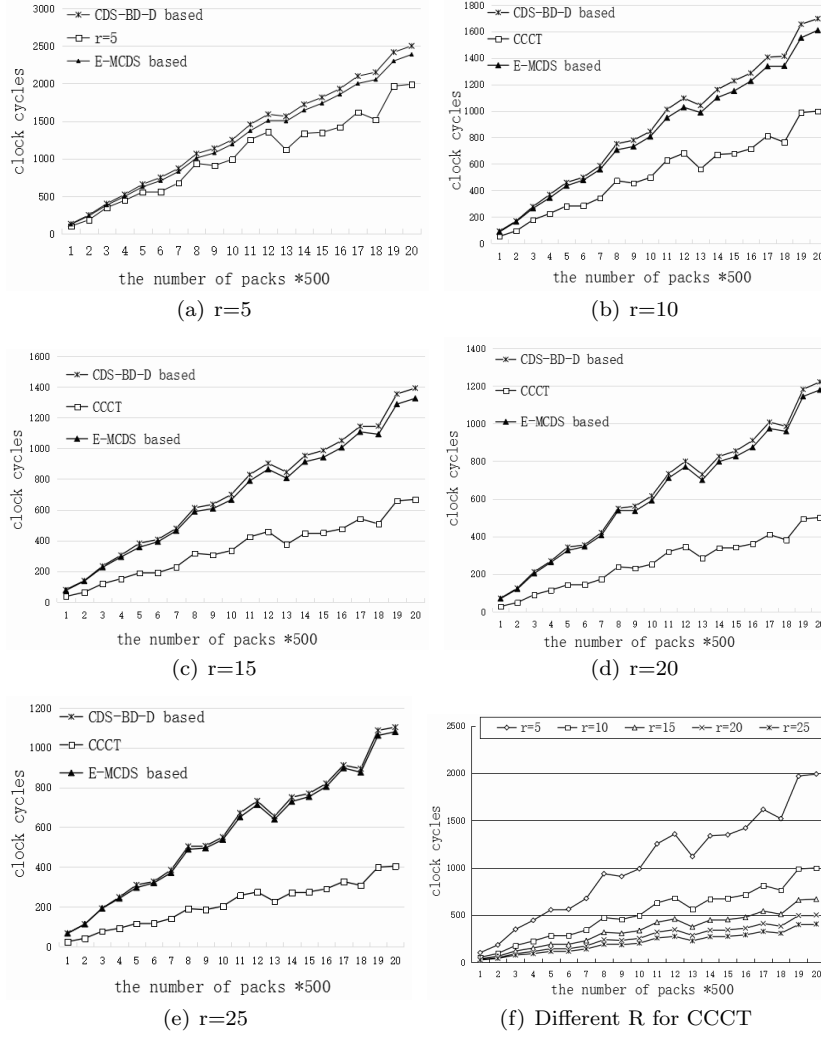


**Fig. 16** Throughput of the CCCT algorithm

### 6.3 Competition of the delay

In simulation tests of delay. Similar to the subsection 6.1, first, we assume 100 sensors spread over some terrain of 100\*100 network. Second, we set the transmission power at one of the following levels 5, 10, 15, 20, and 25. Later, we assume that in each test, there are 500 – 10000 packs need to be forwarded to the sink. When there are more than two packages transmit in the same route, the collision will happen, and only one package can be transmitted. Figs. 17 (a-e) illustrate the clock cycles in which message are transmitted over the number of packs. The cluster head in the CDS transmits the package which is coded by the packs forward from the cluster members. Figs. 17 (a-e) denote the different transmit radios of the sensors when  $r = 5, r = 10, r = 15, r = 20$  and  $r = 25$ , respectively. Each value is the average number by simulating 50 times. It is easy to see that CDS-BD-D based and E-MCDS based algorithms need more clock cycles to transmit the same number of packs than our CCCT algorithm under different transmit radios. The main reason of these results is that there are more collisions in the CDS-BD-D based and E-MCDS based algorithms. However, our CCCT algorithm considers the collision features of the transmit, and is designed by the crossed cube tree. This kind of construction has no collision, which is agree well with Theorem 1 discussed in section 5. Thus, it is more effective when the slot is considered.

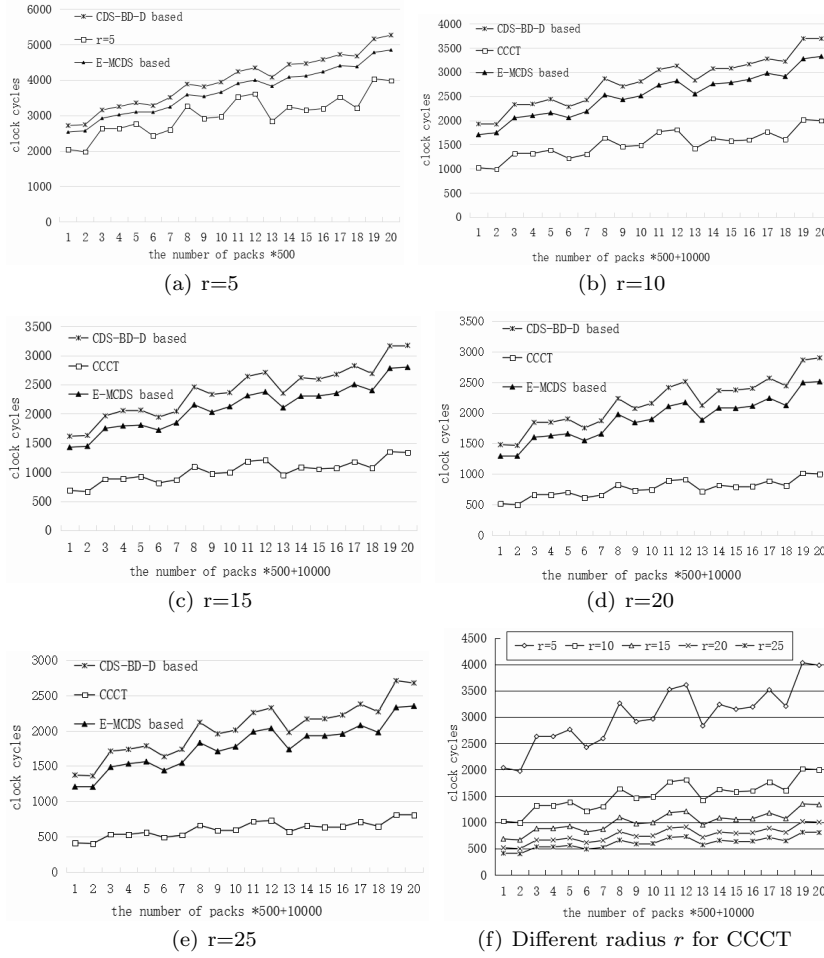
Furthermore, Fig. 17 (a) shows that it improves about 20% and 17% competing to the CDS-BD-D based and E-MCDS base algorithms to our CCCT algorithm when  $r = 5$ , respectively. The improvements are about 40% and 38% when  $r = 10$  (see Fig. 17 (b)), 52% and 50% when  $r = 15$  (see Fig. 17 (c)), 58% and 57% when  $r = 20$  (see Fig. 17 (d)), 63% and 62% when  $r = 25$  (see Fig. 17 (e)), respectively. The lager the radio is, the higher the effectiveness is, and the less clock cycles are, because more packs can be coded into one packet. While comparing to the CCCT, the collision still happens base on the



**Fig. 17** Comparison for delay in 100 sensors

other two algorithms. It means the CCCT is better than the other algorithms. Fig. 17 (f) illustrates the clock cycles in which transmit the message over the different transmit radius. The clock cycles are gradually increasing with the number of packs.

In order to show that the proposed algorithm is also suitable when the number of the packs is higher. Later, we assume 600 sensors spread over the same environment which terrain of  $100 \times 100$  network. We also set the transmission power at one of the following levels 5, 10, 15, 20, and 25. In this simulation, we assume there are 10500-20000 packs needed to forwarded to the sink in each test. Figs. 18(a-e) illustrate the clock cycles in which transmit



**Fig. 18** Comparison for delay in 600 sensors

the message over the number of packs. The same to the Figs. 17, it is easy to see that our algorithm is better than the other algorithms. Fig. 18 (a) shows that it improves about 24% and 18% competing to the CDS-BD-D base and E-MCDS based algorithms to our CCCT algorithm when  $r = 5$ , respectively. The improvements are about 46% and 40% when  $r = 10$  (see Fig. 18 (b)), 57% and 52% when  $r = 15$  (see Fig. 18 (c)), 65% and 60% when  $r = 20$  (see Fig. 18 (d)), 69% and 66% when  $r = 25$  (see Fig. 18 (e)), respectively. The larger the radio is, the higher the effectiveness is and the less clock cycles are because more packs can be coded into one packet. While comparing to the CCCT, the collision still happens base on the other two algorithms. It means the CCCT is better than the other algorithms. Fig. 18 (f) illustrates the clock

cycles in which transmit the message over the different transmit radius. The clock cycles are gradually increasing with the number of packs.

## 7 Conclusion

Data gathering is a fundamental task in Wireless Sensor Networks. Clustering is an effective technique in reducing delay and energy consumption. Currently, most of the existing works focus on constructing Connected Documenting Set based clustering algorithms according to different user requirements. However, due to the hard maintain of the virtual backbone, it is more practical to obtain a CDS construction with small number of clusters. Due to the existence of collision and delay in WSNs, a collision avoid data gathering protocol is needed. To solve the problem, a protocol named Cluster based Converge Cast Tree protocol (CCCT) is presented. Moreover, most literatures neglect the delay factor when constructing Converge Cast Tree. Differently, in this paper, we focus on obtaining a Cluster based Converge Cast Tree (CCCT) under the realistic WSNs. More specifically, three problems are investigated, namely, the Minimal Connected Documenting Set (MCDS) problem, the delay aware CDS problem, and the CCCT construction problem. Later, approximation algorithms and comprehensive theoretical analysis of the approximation factors are presented. Finally, our simulation results show that the proposed algorithms outperform the existing state-of-the-art approaches significantly.

As WSNs interest in the data collecting or tracking events, when there are no update data, the data collection is unnecessary in such place. In the future work, we will focus on when and where the data collection programs are needed. Secondly, WSNs are easily destroyed. The Fault-Tolerance virtual backbone is necessary, when some nodes in the virtual backbone are destroyed. we will focus on how to construct a fault tolerance construction in our future work.

**Acknowledgements** The authors wish to thank National Natural Science Foundation of China (Grant NO: 61072080, 61572010), Natural Science Foundation of Fujian Province of China (2017J05098), The Education Department of Fujian Province science and technology project (JAT160328, JZ160461), and the science research project in Fujian University of Technology (GY-Z160066, GY-Z160130, GY-Z160138).

## References

1. Amodu, O. A., and Mahmood, R. A. R.: Impact of the energy-based and location-based leach secondary cluster aggregation on wsn lifetime, *Wireless Networks*, 1-24 (2016).
2. Ashu1: Ravikant Kaushik, collision detection in WSN through pattern detection and neural network, *International Journal For Technological Research In Engineering*, 2(11): 2538-2541 (2015)
3. Santi P.: Topology control in wireless ad hoc and sensor networks, *Acm Computing Surveys*, 37(2):164-194 (2005).
4. Xu, Y., Zeng, Z. R., and Ding, O.: An energy efficient hole repair node scheduling algorithm for wsn, *Wireless Networks*, 1-14 (2015).

5. He, J., Ji, S., Pan, Y., and Li, Y.: Constructing Load-Balanced Data Aggregation Trees in Probabilistic Wireless Sensor Networks, *IEEE Transactions on Parallel and Distributed Systems*, 25(7): 1681-1690 (2014)
6. Marappan, P., and Rodrigues, P.: An energy efficient routing protocol for correlated data using CL-LEACH in WSN, *Wireless Networks*, 22 (4): 1-9 (2016).
7. Zhao, Y., Wu, J., Li, F. and Lu, S.: On maximizing the lifetime of wireless sensor networks using virtual backbone scheduling, *IEEE Transactions on Parallel and Distributed Systems*, 23(8): 1528-1535 (2012)
8. Zhang, J., Xu, L., Zhou, S., Wu, W., Ye, X.: An efficient connected dominating set algorithm in WSNs based on the induced tree of the crossed cube, *International Journal of Applied Mathematics and Computer Science*, 25(2):295-309 (2015)
9. Sert, S. A., Bagci, H., and Yazici, A.: MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks, *Appl. Soft Computing*, 30, 151-165 (2015)
10. Dong, M., Ota, K., Liu, A., and Guo, M.: Joint Optimization of Lifetime and Transport Delay under Reliability Constraint Wireless Sensor Networks, *IEEE Transactions on Parallel and Distributed Systems*, 27(1): 225-236 (2016)
11. Kim, D., Wu, Y., Li, Y., Zou, F. and Du, D. Z.: Constructing minimum connected dominating sets with bounded diameters in wireless networks, *IEEE Transactions on Parallel and Distributed Systems*, 20(2): 147-157 (2009)
12. Tang, Q., Yang, K., Li, P., Zhang, J., Luo, Y. and Xiong, B.: An energy efficient mcds construction algorithm for wireless sensor networks, *EURASIP Journal on Wireless Communications and Networking*, 1: 1-15 (2012)
13. Cheng, B., Fan, J., and Jia, X.: Dimensional-Permutation-Based Independent Spanning Trees in Bijective Connection Networks, *IEEE Transactions on Parallel and Distributed Systems*, 26(1): 45-53 (2015)
14. Konstantopoulos, C., Vathis, N., and Pantziou, G.: Damianos Gavalas Efficient Delay-constrained Data Collection in Wireless Sensor Networks using Mobile Sinks, *2015 8th IFIP Wireless and Mobile Networking Conference* (2015)
15. Velmani, R. and Kaarthick, B.: An Efficient Cluster-Tree Based Data Collection Scheme for Large Mobile Wireless Sensor Networks, *IEEE SENSORS JOURNAL*, 15(4): 2377-2390 (2015)
16. Doudou, M., Djenouri, D., Barceloordinas, J. M., and Badache, N.: Cost effective node deployment strategy for energy-balanced and delay-efficient data collection in wireless sensor networks, *IEEE Wireless Communications and Networking Conference. "2014 IEEE Wireless Communications and Networking Conference (WCNC)". Istanbul: Institute of Electrical and Electronics Engineers (IEEE)*, 2868-2873, (2014)
17. Cheng, C.-T., Tse, C. K., and Lau, F. C. M.: A delay-aware data collection network structure for wireless sensor networks, *IEEE Sensors J.*, 11(3): 699C710 (2011)
18. Cheng, C.-T., Tse, C. K., and Lau, F. C. M.: A Delay-Aware Network Structure for Wireless Sensor Networks With Consecutive Data Collection Processes, *IEEE Sensors J.*, 13(6): 2413-2422 (2013)
19. Khan, S., Khan, F., and Khan, S.: Delay and Throughput Performance Improvement in Wireless Sensor and Actor Networks, *The 5th IEEE National Symposium on Information Technology: Towards Smart World*, 1-5 (2015)
20. Xu, X. H., Li, M., Mao, X. F., Tang, S. J., and Wang, S. G.: A delay efficient algorithm for data aggregation in multihop wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst.*, 22(1): 163-175 (2011)
21. Li, H. X., Wu, C., Hua, Q. S., and Lau, F. C. M.: Latency-minimizing data aggregation in wireless sensor networks under physical interference model, *Ad Hoc Network*, 12(1): 52-68 (2014)
22. Chen, I., and Wang, Y.: Reliability Analysis of Wireless Sensor Networks with Distributed Code Attestation, *IEEE Communications Letters*, 16(10): 1640-1643 (2012)
23. Sidera, A., and Toupis, S.: On the delay/cost tradeoff in wireless mobile Delay-Tolerant Networks, *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. IEEE*, 452-459 (2014)
24. Hong, C., Xiong, Z., and Zhang, Y.: A hybrid beaconless geographic routing for different packets in WSN, *Springer-Verlag New York, Inc.* 22:1107-1120 (2016)
25. Ahlswede, R., Cai, N., Li, S. Y. R., and Yeung, R. W.: Network information flow, *IEEE Transactions on Information Theory*, 46(4): 1204-1216 (2000)

26. Ye, X., Li, J., and Xu, L.: Distributed separate coding for continuous data collection in wireless sensor networks, *Acm Transactions on Sensor Networks*, 11(11): 1-26 (2014)
27. Gadouleau, M., Richard, A., and Fanchon, E.: Reduction and Fixed Points of Boolean Networks and Linear Network Coding Solvability, *IEEE Transactions on Information Theory*, 62(5):2504-2519 (2016)
28. Xu, L., Zhang, J., Xiang, Y., and Huang, X.: Network coding based converge-cast scheme in wireless sensor networks, *Wireless Personal Communications*, 96(4): 4947-4972 (2017)
29. Cheng, B., Fan, J., and Jia, X.: Dimensional-permutation-based independent spanning trees in bijective connection networks, *IEEE Transactions on Parallel and Distributed Systems*, 26(1): 45-53 (2014)
30. Zhang, J., Xu, L., Zhou, S., Min, G., Xiang, Y., and Hu, J.: Crossed cube ring: a k-connected virtual backbone for wireless sensor networks, *Journal of Network and Computer Applications*, 91(1): 75-88 (2017)